NUI Galway
OÉ Gaillimh

Autumn Examinations 2014

Exam Code(s)	4BCT, 3BA, 1SD, 1MDM
Exam(s)	4th Year B.Sc. (CS&IT) 3rd Year B.A. (Information Technology) Higher Diploma in Applied Science (Software Design & Development) Masters in Digital Media
Module Code(s)	CT404, CT336
Module(s)	Graphics and Image Processing
Paper No.	
Repeat Paper	Yes
External Examiner(s)	Prof. L. Maguire
Internal Examiner(s)	Prof. G. Lyons Dr. M. Madden * Dr. S. Redfern

Instructions: Answer any three questions.
All questions carry equal marks.

Duration	2 hours
No. of Pages	6
Discipline(s)	Information Technology
Course Co-ordinator(s)	

Requirements:
MCQ
Handout
Statistical/ Log Tables
Cambridge Tables
Graph Paper
Log Graph Paper
Other Materials
Graphic material in colour Yes ☒ No □

Release to Library: Yes ☒ No □

PTO

Q.1. (Graphics)

(i) What is a surface normal? [2]

Discuss, with the use of a diagram in each case, the following 3D graphics techniques, and identify the importance of surface normals to each

- Flat (Lambert) Shading [2]
- Normal Interpolating (Phong) Shading [2]
- Bump Mapping [2]
- Back-Face Culling [2]

(ii)
Explain in general the keyframe approach to animation in computer graphics, and explain specifically its use in X3D, referring to the TimeSensor, Transform, OrientationInterpolator and PositionInterpolator nodes in your answer. [5]

(iii)
Consider the HTML5/Javascript code below. This code handles keypresses on the document by invoking the `draw()` function. The key that was pressed is sent as a single-character string. Write suitable Javascript code to manipulate the Canvas context object in order to:

- Clear any existing graphics [1]
- Draw a rectangle whose position changes by 1 pixel each time one of these four keys are pressed: (W=up, A=left, D=right, S=down) [4]

```
<html>
 <head>
  <script>
        function attachEvents() {
               document.onkeypress = function(event) {
                  var key = String.fromCharCode(event.keyCode || event.charCode);
                  draw(key);
               }
        }
        function draw(key) {
          var canvas = document.getElementById("canvas");
          var context = canvas.getContext('2d');

          // to do: clear existing graphics on the Canvas

          // to do: draw a rectangle whose position changes based on key
          // that was pressed (W=up, A=left, D=right, S=down)

        }
  </script>
 </head>
 <body onload="attachEvents();">
   <canvas id="canvas" width="300" height="300"></canvas>
 </body>
</html>
```

Q.2. (Graphics)

(i) *The Z-Buffer Algorithm*

- With regard to the rendering of 3D graphics, discuss in detail the operation of the Z-Buffer Algorithm. [3]
- Explain what it means for this algorithm to be an 'image space' technique. [2]
- How might the Z-Buffer be useful for post-rendering effects such as depth blur? [2]

(ii) Consider the display of a realistic forest in an interactive 3D graphics environment: the key trade-off is frame-rate versus polygon count. In this context, discuss the use of textures, visibility culling, levels of detail (LODs), mipmaps, bumpmaps and billboards in order to obtain a maximal frame-rate. Illustrate your answer with diagrams. [10]

(iii) Why are *back buffers* used in real-time graphics applications? Explain their operation. [3]

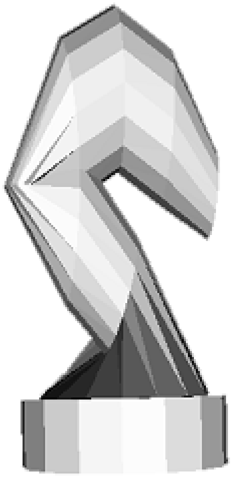
Q.3. (Graphics)

(i) Discuss how graphics hardware has changed (and improved) over the past decade. In your answer, refer to these terms: Graphics Processing Unit (GPU), on-board RAM, texture memory, data bus, DirectX/OpenGL [5]

(ii) Why are nested co-ordinate systems useful for 3D graphics/animation programming? In your answer, explain and provide code samples illustrating the use of nested co-ordinate systems in both Canvase/HTML5 *and* X3D. [5]

(iii) Describe the use of extrusion in X3D, referring to each of the seven fields used by the Extrusion node (see final page of exam paper for these fields). [5]

(iv) Write X3D code to produce an extruded shape, similar to the chess piece shown below. You should consider its geometry only, and can ignore the use of materials. Note that the most useful X3D nodes, are summarised on the final page of this exam paper. [5]



Q.4. (Image Processing)

(i)

Convolution (also called ‘filtering’) is one of the most fundamental low-level image processing techniques.

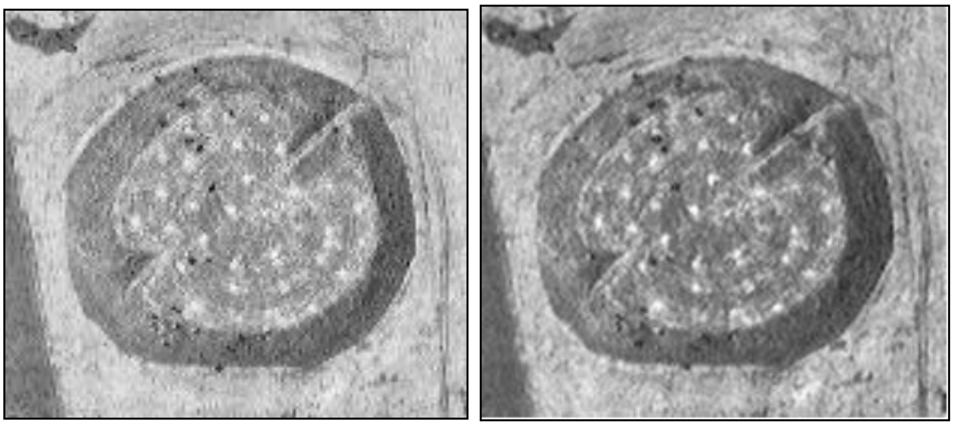
- Describe, with use of a diagram, the convolution algorithm. [3]
- Present a convolution kernel suitable for noise reduction. Explain how this kernel produces the desired result, assuming a greyscale image with pixel values in the range 0 to 255. [3]
- Present a convolution kernel suitable for edge detection. Explain how this kernel produces the desired result, assuming a greyscale image with pixel values in the range 0 to 255. [4]

(ii)

With regard to the use of image processing for the capturing of 3 Dimensional (3D) information, define the terms: (i) active depth sensing, (ii) passive depth sensing.

[3]

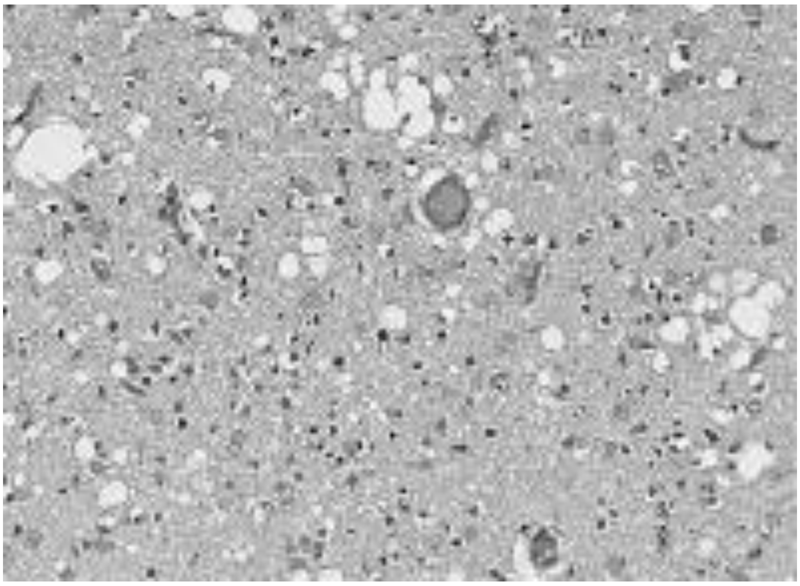
Outline, at a high level, a suitable algorithm for extracting 3D information from a stereo pair of images, such as the image pair illustrated below. [7]



Q.5. (Image Processing)

(i) Describe the *mathematical morphology* approach to image processing. Outline some typical circumstances in which this approach is useful [8]

(ii) The image below is taken from tissue sample of a human brain affected by neurological damage. Of interest are white areas that are at least 5 pixels in diameter. **Devise** and **discuss** an algorithm for **automatic** and **robust** isolation of areas matching this specification [12]



Some useful X3D nodes:

Node	Important Fields and Nested Nodes
Shape	Nested Nodes: Appearance, Geometry Nodes (Box, Sphere, Cone, Cylinder, Text, Extrusion, etc.)
Appearance	Nested Nodes: Material, ImageTexture
Material	Fields: diffuseColor, specularColor, emissiveColor, ambientIntensity, transparency, shininess
ImageTexture	Fields: url
Transform	Fields: translation, rotation, scale, center. Nested Nodes: Other Transforms, Shapes, Sensors
TimeSensor	Fields: enabled, startTime, stopTime, cycleInterval, loop
PositionInterpolator	Fields: key, keyValue
OrientationInterpolator	Fields: key, keyValue
Extrusion	Fields: crossSection, spine, scale, orientation, beginCap, endCap, creaseAngle
Box	Fields: size
Sphere	Fields: radius
Cylinder	Fields: radius, height, side, top, bottom
Cone	Fields: height, bottomRadius, side, bottom
PointLight	Fields: on, location, radius, intensity, ambientIntensity, color, attenuation
ROUTE	Fields: fromNode, fromField, toNode, toField

Some useful methods/properties of the Canvas 2D Context object:

Method/Property	Arguments/Values	Notes
fillRect	(Left, Top, Width, Height)	Draw a filled rectangle
beginPath	None	Start a stroked path
moveTo	(X, Y)	Move the graphics cursor
lineTo	(X, Y)	Draw a line from graphics cursor
stroke	None	End a stroked path
fillStyle	="rgb(R,G,B)"	Set fill colour
strokeStyle	="rgb(R,G,B)"	Set line colour
save	None	Save the current coordinate system
restore	None	Restore the last saved coord system
translate	(X,Y)	Translate the coordinate system
rotate	(angle)	Rotate the coordinate system clockwise, with angle in radians
scale	(X,Y)	Scale the coordinate system independently on the X and Y axes